Université Echahid Hamma Lakhdar **El-Oued le : 14/01/2024**
Département d'Informatique **2ème LMD Informatique**
Module : Algorithmique et Structures de Données **Durée : 1h :30**

Correction de l'examen du troisième semestre

Exercice 01:

Soit une pile **P** contenant des chaînes de caractères, tel que :

```
char  Mot[10], Mot1[10];
      EMPILER (P, "le");
      EMPILER (P, "ferme");
      EMPILER (P, "la");
      DEPILER (P, Mot);
      printf("%s ", Mot);
      EMPILER (P, "voile");
      EMPILER (P, "belle");
      EMPILER (P, "rapidement");
      DEPILER (P, Mot);
      DEPILER (P, Mot);
      printf("%s ", Mot);
      DEPILER (P, Mot);
      DEPILER (P, Mot1);
      printf("%s ", Mot1);
      DEPILER (P, Mot1);
      printf("%s ", Mot1);
      printf("%s.\n", Mot);
```

1- Qu'est-ce qui sera affiché à la fin ce bout de programme ?

Solution : « la belle ferme le voile »

Exercice 02 :

Soit la fonction main suivante :

```
void main(){
int min,max,n ;
int a[8]={10, 12, 14, 16, 18, 2, 7, 25} ;
MinMaxTab(a , &min, &max, 8);
printf("le minimum est= %d\n", min);
printf("le maximum est = %d\n", max);}
```

1- Ecrire en langage C la procédure **`void MinMaxTab (int T[], int* min, int* max, int n)`** qui permet de calculer le minimum et le maximum d'un tableau T ; les deux variables min et max sont deux pointeurs et n est la taille du tableau.

Solution :

```
#include <stdio.h>
void MinMaxTab (int t[], int* min, int* max, int n)
{
int minimum=t[0],maximum=t[0],i;
for(i=0;i<n;i++)
{
```

BON COURAGE

Correction de l'examen du troisième semestre

```
if(minimum>=t[i])
{
minimum=t[i];
}
if(maximum<=t[i])
{
maximum=t[i];
}
}
*min=minimum;
*max=maximum;
}
void main()
{
int min,max,n ;
int a[8]={10, 12, 14, 16, 18, 2, 7, 25} ;
MinMaxTab(a , &min, &max, 8);
printf("le minimum est= %d\n", min);
printf("le maximum est = %d\n", max);
}
```

Exercice 03 :

1- Ecrire une fonction : **`Liste *ajouterEnDebut(Liste *L, int valeur)`**, qui permet d'ajouter un element en debut d' une liste simple L et donne une nouvelle tête de liste tel que :

```
typedef struct Liste {
int val;
struct Liste *suivant;
}Liste;
```

Solution :

```
Liste *ajouterEnDebut(Liste *L, int valeur)
{
    Liste * nouveau=malloc(sizeof(Liste))    ;
    nouveau->val=valeur;
    nouveau->suivant=L;
    return nouveau;}
```

BON COURAGE

Correction de l'examen du troisième semestre

2- Ecrire une deuxiéme fonction : **`Liste *viderListe(Liste *L)`** qui permet de vider une liste L et retourne une liste vide.

```
Liste *viderListe(Liste *L){
        Liste* tmp;
        if(L==NULL)
        printf("echec:la liste est vide");

        while(L!=NULL)
        {       tmp=L;
                L=L->suivant;
                free(tmp);
        }
        if(L==NULL)
        printf("la liste devenue vide\n");
    return  L;
}
```

BON COURAGE